Amendments to the Specification:

Please replace the application title at page 1, line 1 with the following amended title:

SYSTEM AND METHOD FOR SNAPSHOT STORAGE MANAGEMENT WITH INDEXING AND USER INTERFACE ~~MANAGING STORED DATA~~

Please replace paragraph 28 with the following amended paragraph:

[00028] Fig. 9 is a block diagram depicting a storage system ~~flow diagram illustrating a method for producing a copy of a primary volume~~ according to an embodiment of the invention.

Please replace paragraph 29 with the following amended paragraph:

[00029] Fig. 10 is a flow diagram illustrating a method for producing a copy of a primary volume ~~block diagram depicting a storage system~~ according to an embodiment of the invention.

Please replace paragraph 33 with the following amended paragraph:

[00033] A primary data set generally denotes a volume, application data, or other data being actively used by a client computer. A volume is generally an area of storage on a data storage device, which may be the whole storage area or portions thereof. An agent 102 generally refers to a program module that provides control and data transfer functionality to client computers. A client generally refers to a computer with data that may be stored, such as a personal computer, workstation, mainframe computer, host, etc. An application[[,]] is software that utilizes data to be stored such as WINDOWS 2000

File System, MICROSOFT EXCHANGE 2000 Database, etc. Intelligent data agent 104 is an agent for an application that provides control and data transfer functionality for the data of the specific application. A plurality of agents 102, such as intelligent data agents 104 or quick recovery agents 108, may be provided for and/or reside on each client computer, for example, where the client computer includes a plurality of applications and a file system or systems for which a quick recovery volume may be created.

Please replace paragraph 45 with the following amended paragraph:

[00045] The file system layer 210 represents a logical view of the data, which typically includes of a hierarchy of nested directories, folders, and files, and metadata. The features and attributes of files may vary according to the particular file system in use. For example, an NTFS, i.e., a WINDOWS NT file system, tracks ownership and per-user access rights on each file, whereas FAT, i.e., file allocation table, file systems do not provide security features. The applications layer 204 includes application software, such as a word processor program, etc. which interface with the file system provided by the operating system to store data. Sophisticated software, such as database management systems ("DBMS"), may use special file system features or even raw logical volumes, ~~and employ or measure~~ to protect the consistency of data and metadata. The consistency of the data and metadata may be maintained during the creation of a quick recovery volume with writers particular to specific applications or file systems. Writers are described in more detail below.

Please replace paragraph 47 with the following amended paragraph:

[00047] In the snapshot phase 408, quick recovery agent 108 synchronizes with the application to be stored, if any, and the operating system to ensure that all data of the primary data set to be stored is flushed to the archival storage unit or destination disk. In these disks, the quick recovery volume of the primary data set will be stored. The snapshot phase 408 also ensures that the primary volume 114 where the primary data set, such as the primary volume or application data, is located is not modified during the creation of the snapshot image. This may be accomplished by suspending input or output to the primary volume 114 containing the primary data set, step 420, which will ensure that the file system and metadata remain unchanged during the copy operation. Quick recovery agent 108 evokes the snapshot mechanism to create a snapshot image of the primary data set, step 422. The snapshot mechanism or snapshot manager may be a software module, an external snapshot application, such as ~~COMMVALUT~~ COMMVAULT SOFTWARE SNAPSHOT, XP/.NET, TIMEFINDER, etc., or a combination thereof. Once the snapshot image of the primary volume 114 is created, application access to primary disk resumes, step 424, and update data on the primary volume 114 as necessary, while the copy operation for the quick recovery volume 118 is in progress or is pending. Once the snapshot image is made the snapshot image is indexed, step 425. Indexing generally denotes associating information with a snapshot image that may be useful in managing the snapshot image, such as the date the snapshot image was created, the lifespan of the snapshot, etc.

Please replace paragraph 59 with the following amended paragraph:

[00059] In each snapshot a user may drill down to view particular folders, files, etc., or to view particular objects native to applications. Users are able to specify, with regard to a quick recovery copy, (1) whether the snapshot image should persist after the creation of the quick recovery volume, (2) if the image should persist, for how long, and (3) the location of the persistent storage for the image. Users are able to (1) request a snapshot image and a quick recovery volume, just a snapshot image, or just a quick recovery volume, (2) request a software snapshot image and optionally specify that it be converted to a hardware snapshot image, (3) request that the hardware snapshot image persist for a certain period of time, (4) recover data from a snapshot image at the volume level, e.g., the whole volume, or sub-volume level, e.g., individual folders, files, objects, etc., and (5) make another copy of a snapshot image on the SAN.

Please replace paragraph 65 with the following amended paragraph:

[00065] The invention provides a user interface screen so that a user may browse and recover data, such as from snapshot images, quick recovery volumes, primary copies, storage copies, etc., as of a point in time. Browsing and recovery may be by client, sub-client, volume, and application specific, and may be at the volume level or at the sub-volume level. Volume level recovery refers to replication of entire volumes, whereas sub-volume level refers to recovery at a folder, file, or object level. Referring to Fig. 6, there is shown a browser interface screen 600, according to the invention. Screen 600 includes a plurality of frames, such as directory frame 602 and a contents frame 604. The directory frame generally provides a list of all available drives, partitions, volumes,

snapshots, storages, etc. and the file folders therein, of a client computer in a hierarchical arrangement. The contents frame 604 generally lists the contents of any item appearing in the directory frame 602, such as folders, files, or objects. The contents may be displayed by highlighting any one of the items in the directory frame 602. By selecting the "My Snapshots" folder 610, for example, the contents 612 of the snapshots folder 610 are displayed in the contents frame 604. The contents may be displayed with relevant details, such as the date of creation, persistence, association, the capacity of the volume, etc. A user may change the properties of a snapshot, such as how long a particular snapshot will persist, the location, etc., and the user may direct the creation of another volume or copy of a software snapshot using, for example, ~~COMMVALUT~~ COMMVAULT data movers.

Please replace paragraph 68 with the following amended paragraph:

[00068] Referring to Fig. 7, there is shown browser interface screen 700. Screen 700 allows for browsing snapshot images according to the invention and displays a particular client 702 as a folder, for example, a folder for the client squid.commvault.com. At least one subfolder 704 may be displayed showing the application or applications available with respect to the client for creating storage copies, browsing, and recovery; for example, "Exchange 2000 Database," "File System," and "SQL Server 2000." Selecting one of the subfolders, such as "Exchange 2000 Database" reveals further subfolders therein, such as a "SnapShot Data" subfolder 706, which provides snapshot data for the selected application. Further drilling down through the subfolders will cause to be displayed in a hierarchical layout the snapshots available, such

as "SnapShot 1" 708,[["]] application objects 710, such as the "Information Store," "First Storage Group", "Mailbox Store", "Public Folder Store", etc. A user may then perform a single click recovery or restore, or creation of a snapshot image or quick recovery volume of a primary volume or application data, or of any storage copy, by selecting the level from which the data displayed thereunder will be stored or recovered. For example, by selecting "First Storage Group" and right clicking, the user will be presented with an activity window 712, which allows the user to select the "Recover" function. In this instance, by selecting "Recover" all data related to the objects appearing under the "First Storage Group" will be recovered. The user may choose to recover a single file, or a single object, such as a single EXCHANGE store or SQL Server database. The restoration may be implemented with a fast data mover, such as COMMVAULT'S DATAPIPE described in detail in Application No. 09/038,440, which moves data from disk to disk. Additionally a disk-to-disk server-less data mover can be implemented as well.

Please replace paragraph 70 with the following amended paragraph:

[00070] The user interface screen may also be used to request destruction of a given snapshot copy on a single item ~~bases~~ basis or automatically upon the lapse of the persistence period. Basic information for each snapshot copy destroyed, or otherwise, may be stored for purposes of tracking and display. This information may be stored in an MSDE database, but can also be stored in other similar data structures.

Please replace paragraph 81 with the following amended paragraph:

[00081] Fig. ~~9~~10 is a flow diagram showing a method of performing raw partition incremental secondary storage operations according to an embodiment of the invention. Referring to Figs. 9 and 10, q~~Q~~uick recovery volume 118 preserves not only the data of primary volume 114, but also how applications and other processes recognize and interact with that data. For example, the system preserves the logical identifiers used by applications to recognize volumes. System 100a also conveys and preserves application-level data by, among other techniques, copying data at the block level when taking a snapshot and not copying at merely the file level. Blocks of data captured in a snapshot are thus captured and copied as they are laid out on the disk or other storage media.

Please replace paragraph 84 with the following amended paragraph:

[00084] Still r~~R~~eferring to Figs. ~~8~~9 and ~~9~~10, system 100a captures a snapshot of primary volume 114, step ~~800~~950. As previously described, the snapshot contains a block level copy of the data on primary volume 114. Intelligent data agent 104, which can be aided by the copy manager 116, packages the snapshot into a backup format or other format and copies the snapshot to secondary storage 120, step ~~802~~955. For example, intelligent data agent 104 may package a snapshot according to GALAXY'S format or another backup format or secondary storage format as further described in Patent No. 5,559,991, Patent No. 5,642,496, and Patent No. 6,418,478 all of which are hereby incorporated herein by reference in their entirety. Intelligent data agent 104 packages additional information with the snapshot such as information regarding how applications recognize the snapshot data, where snapshot data is physically located on the

primary volume 114, where snapshot data is logically located on the primary volume 114, application-specific preferences associated with the snapshot data, and other information. The additional information can be included, for example, in header information of the snapshot data being copied to secondary storage 120, in the payload of the snapshot data being copied to secondary storage 120, or in combinations thereof. Intelligent data agent 104 may avoid packaging the snapshot into a backup format, and instead copy the snapshot in its original format to secondary storage 120, for example, in the case of creating a quick recovery volume in secondary storage.

Please replace paragraph 85 with the following amended paragraph:

[00085] Block filter 117 monitors the primary volume 114 for changes, step ~~804~~960. The block filter 117 compares the blocks of primary volume 114 against information stored in cache 119 regarding the state of the blocks of primary volume 114 at the time of the previous snapshot. If the state of a block is different on the primary volume 114 from the state of the block recorded in the cache 119, then block filter 117 indicates that a change has occurred to the block on primary volume 114. As previously described, block filter 117 is a software module or other daemon that monitors the primary volume 114 in real time. Block filter 117 checks the primary volume 114 for changes as directed by a user or according to a scheduled frequency such as a frequency specified by a storage policy. Block filter 117 also monitors for incremental block level changes that occurred since a previous snapshot.

Please replace paragraph 86 with the following amended paragraph:

[00086] Turning to Fig. 11 block diagram is shown depicting incremental block level changes occurring in a series of raw partition storage operations according to an embodiment of the invention. Fig. 11 shows a first storage operation 900, a second storage operation 902, a third storage operation 904, a fourth storage operation 905~~5~~6, a fifth storage operation 908, a sixth storage operation 910, and a number of blocks 912 being copied. First storage operation 900 is a full snapshot of a raw partition primary volume 114 comprising six blocks 912 A_0, B_0, C_0, D_0, E_0, and F_0. First storage operation 900 stores a snapshot of the blocks 912 in their original or base state. Block filter 117 monitors primary volume 114 for changes that occur to blocks 912 since the first full snapshot 900 was made.

Please replace paragraph 87 with the following amended paragraph:

[00087] For example, second storage operation 902 is an incremental snapshot of the primary volume blocks that changed since first storage operation 900. An incremental storage operation or snapshot stores only copies of those elements of the volume that have changed since a previous storage operation. As shown, only A_0, B_0, and E_0 have changed since first storage operation 900, and so system 100 takes an incremental snapshot and stores only the changed blocks A_1, B_1, and E_1. Third storage operation 904 is also an incremental snapshot and stores copies of only those blocks that changed since second storage operation 902. Thus, C_1 and E_2 are stored. Fourth storage operation 906 is another incremental snapshot and causes copies of blocks A_2 and D_1.

Fifth (n-1) storage operation 908 is another incremental snapshot and causes a copy of block A_3.

Please replace paragraph 88 with the following amended paragraph:

[00088] Storage operations continue until another full snapshot is taken. Sixth (n) storage operation 910 presents another full snapshot of the raw partition primary volume 114. Sixth (n) storage operation 910 stores a full snapshot of the blocks 912 capturing the changes that occurred to all the blocks 912 since first storage operation 900. Sixth (n) storage operation 910 may be made as a synthetic storage operation. A synthetic storage operation takes a previous full storage operation of a volume and incorporates changes to the previous full storage operation recorded in one or more incremental storage operations to create an updated full storage operation without using the system resources that would be required to make a second storage operation directly from the volume itself. For example, here sixth (n) storage operation 910 is a full snapshot of the first volume 114 created by combining the most recent change to A_0 stored as A_3 in fifth storage operation 908, the most recent change to B_0 stored as B_1 in second storage operation 902, the most recent change to C_0 stored as C_1 in third storage operation 904, the most recent change to D_0 stored as D_1 in fourth storage operation 906, the most recent change to E_0 stored as E_2 on the third storage operation 904, and F_0 (which has not changed) from first storage operation 908. Since the sixth (n) storage operation 910 is a full snapshot of the first volume 114 created via synthetic copy, the system does not need to take another full snapshot of the first volume 114.

Please replace paragraph 89 with the following amended paragraph:

[00089] Returning to Fig. ~~8~~10, when the block filter 117 detects a change to one or more blocks on the primary volume 114, step ~~806~~965, the system performs an incremental storage operation of any changed blocks and copies only the changed blocks to secondary storage 120, step ~~808~~970.

Please replace paragraph 91 with the following amended paragraph:

[00091] Block filter 117 queries storage manager 110 or other controlling process or policy whether it should continue to monitor block level changes to primary volume 114, step ~~810~~975. Control either returns to step 804 and block filter 117 continues to monitor changes, or the routine exits, step ~~812~~980.

Please replace paragraph 92 with the following amended paragraph:

[00092] Snapshot data and incremental block level changes to snapshot data may be copied to remote secondary storage, 120. For example, system 100a may be integrated with a company's remote disaster recovery ~~DR~~ strategy since incremental block level changes can be copied thereby reducing network overhead generally associated with remote disaster recovery ~~DR~~ operations. Intelligent data agent 104, which can be aided by the copy manager 116, may package the snapshot data and incremental block level changes to the snapshot data into a backup format, and back up the data over a network to remote secondary storage 120. System 100a uses transport protocols known in the art such as TCP/IP, Ethernet, Gigabit Ethernet, and other protocols. System 100a may use the COMMVAULT DATAPIPE further described in

Patent No. 6,418,478 which is incorporated herein by reference in its entirety. System 100a may use additional information communicated with the snapshot data and block level incremental changes to snapshot data such as information regarding how applications recognize the data, where the data is physically located on the primary volume 114, where the data is logically located on the primary volume 114, application-specific preferences associated with the data, and other information, to unpack the data at the remote secondary storage 120 location and to create a remote quick recovery volume 118.